Eurographics / IEEE Symposium on Visualization 2011 (EuroVis 2011)
H. Hauser, H. Pfister, and J. J. van Wijk
(Guest Editors)

Volume 30 (2011), Number 3

Depth of Field Effects for Interactive Direct Volume Rendering

Mathias Schott[1,2], A. V. Pascal Grosset[1,2], Tobias Martin[2], Vincent Pegoraro[3], Sean T. Smith [4,5], Charles D. Hansen[1,2]

[1]SCI Institute, University of Utah, USA, [2]School of Computing, University of Utah, USA, [3]Saarland University, Germany,
[4]Department of Chemical Engineering, University of Utah, USA, [5]Institute for Clean and Secure Energy, University of Utah, USA

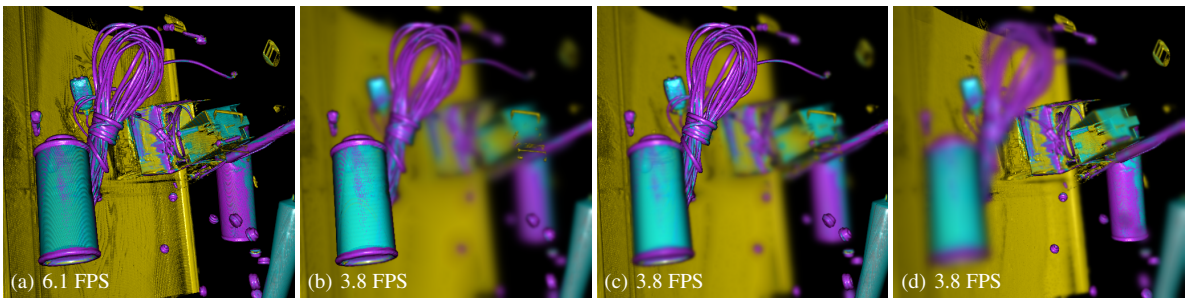


Figure 1: *The backpack data set with* $512 \times 512 \times 373$ *voxels rendered a) with Phong shading only; and b) with depth of field with* $\alpha = 30.9°$ *focused on the spray can in the foreground, c) on the wires behind it and d) on the boxes with the other spray can in the background. In each image 1469 slices were taken and gradients were estimated on the fly.*

Abstract

In this paper, a method for interactive direct volume rendering is proposed for computing depth of field effects, which previously were shown to aid observers in depth and size perception of synthetically generated images. The presented technique extends those benefits to volume rendering visualizations of 3D scalar fields from CT/MRI scanners or numerical simulations. It is based on incremental filtering and as such does not depend on any pre-computation, thus allowing interactive explorations of volumetric data sets via on-the-fly editing of the shading model parameters or (multi-dimensional) transfer functions.

Categories and Subject Descriptors (according to ACM CCS): Computer Graphics [I.3.7]: Three-Dimensional Graphics and Realism— Subjects: Color, shading, shadowing, and texture

1. Introduction

Volumetric rendering is widely used to visualize 3D scalar fields from CT/MRI scanners and numerical simulation data sets. An important aspect of volumetric rendering is the ability to provide perceptual cues to aid in understanding of structures contained in the data.

Recently, a perceptual model [HCOB10] has been proposed that explains how the human visual system uses the depth of field introduced by the lens of the eye to aid inferring absolute distances. This model allows one to manipulate the conveyed scale of synthetically generated images, and as such can be used to improve the effectiveness of scientific visualizations, by emphasizing different scales.

Depth of field effects have been researched in the fields of realistic image synthesis where stochastic sampling is often used to generate high quality images, at the cost of considerable computation time. In contrast, there exists a large body of interactive techniques that allow the approximation of depth of field effects for scenes of polygonal geometry,

whereas depth of field effects for volumetric rendering have not received a similar attention.

In this paper, we present a method for interactive direct volume rendering that allows the computation of depth of field effects for volumetric data sets with both solid and transparent features. The proposed method is based on incremental filtering, which has been successfully used in the past to approximate integration for computing advanced scattering and occlusion effects [KPH*03, SPH*09, vPBV10].

Depth of field effects can be easily integrated into a slice-based direct volume rendering system using an incremental filtering approach, which does not require any precomputation, thus allowing interactive examinations of volumetric data sets via on-the-fly editing of the shading model parameters or (multi-dimensional) transfer functions, enabling the classification of voxels in the data set as solid or semi-transparent.

This paper is structured as follows: Section 2 discusses methods related to depth of field and related Focus+Context techniques. Section 3 discusses depth of field and motivates and outlines an integration into a slice-based volume renderer. Results are presented and discussed in Section 4, followed by conclusions and future work in Section 5.

2. Related Work

Depth of field (DOF) is common in photography and is often simulated in ray tracing to enhance realism. In visualization, it has been mainly used to improve depth perception by blurring objects that would appear out of focus when viewed with a physical camera or to guide users to salient regions by blurring less relevant regions in Focus+Context visualizations.

Kosara *et al.* [KMH01, Kos01] proposed to blur out objects which are not currently relevant in a scene by applying a box filter using texture mapping hardware. From the user study [KMH*02] they subsequently carried out, they found that participants were able to locate objects based on those visualizations similarly compared to using color alone or both together, indicating that blurring can convey further information useful for the comprehension of a specific visualization. Mather *et al.* [MS02] report that blur on its own can give an immediate sense of ordering, but observers find it hard to differentiate between different blur levels to give an exact perception of depth. Blur, when combined with other depth cues [MS04], however is a very useful addition to enhance depth perception.

Other Focus+Context techniques are also often used in volume rendering. Viola *et al.* [VKG04] investigated altering the opacity to create cut-away views where opacity and color are modulated to attract the user's gaze to the point of interest. Wang *et al.* [WZMK05] used the optical properties of a lens to magnify the region of interest to see it in greater detail, while deemphasizing the surrounding region which gets slightly distorted and blurry as a result. Along the same lines, Wang *et al.* [WWLM11] use a grid-based approach to enlarge the region of interest at the expense of the other regions. Another Focus+Context technique proposed by Rautek *et al.* [RBGV08] uses fuzzy logic to determine where the user is focusing in a volume, and puts emphasis there by progressively decreasing the transparency between regions in focus and out of focus.

Ropinski *et al.* [RSH06] evaluated DOF among a number of other techniques to help depth perception in angiography where they found it to be useful to improve depth perception. Ciuffreda *et al.* [CWV07] present an empirically based, conceptual model of human blur perception and its impact on the blur-based depth-ordering of objects. Held *et al.* [HCOB10] present a probabilistic model of blur, introduced by the lens of the human eye, which can be used by a viewer together with further relative depth information to give cues about absolute distances in a scene. Their model was validated by a user study where subjects had to estimate absolute distances of objects in synthetic polygonal scenes. Our proposed depth of field method can be used achieve similar effects for visualizations of volumetric data sets.

A short survey of various depth of field techniques has been compiled by Demers [Dem04] and a detailed survey of further depth of field techniques has been compiled by Barsky *et al.* [BK08]. Many of these approaches are image-space techniques which blur an initially generated in-focus image. Potmesil and Chakravarty [PC81] use linear filtering in a post-processing stage to adaptively blur images according to their distance from the focal plane. This technique is fast, since it does not attempt to minimize depth discontinuities or color bleeding. Distributed ray tracing has also been used to compute depth of field effects [CPC84, NSG90], however many interactive approximations for computing depth of field effects interactively are not based on ray tracing, due to the high performance cost.

The layered depth of field presented by Scofield [Sco92] and referenced in other work [Dem04, BK08] is particularly relevant since our proposed depth of field method uses the intrinsic layers of a slice-based direct volume renderer. This technique applies one blur level per layer which causes bands between the different layers. Barsky *et al.* [BHK*03a, BHK*03b] improve this technique by rendering scenes consisting of opaque geometry into several layers and blurring them individually, which Kraus *et al.* implement on the GPU [KS07]. Our proposed method similarly performs blurring of composited images inherent to the slice traversal of a direct volume rendering system. This incremental blurring, shown by Rokita *et al.* [Rok96] to be equivalent to a larger effective blur, reduces aliasing artifacts due to undersampling.

Depth of field has also been investigated for volume rendering by Crassin *et al.* [CNL*09, CNLE09] to render large

out-of-core data sets. They compute an object space sparse voxel tree and use it to render massive data sets via a view-dependent level-of-detail approach which can also be used to approximate depth of field effects by performing mipmap level lookups into their data structure. They note that animation is an issue with their approach, indicating that their preprocessing is rather expensive, though details are not given. The method proposed in this paper does not depend on precomputation and as such allows on-the-fly replacement of the volume data, e.g. by paging in volumes of a time series.

The method proposed by Ropinski *et al.* [RSH06] computes a depth buffer based on opacities accumulated along rays while marching through a volume. The depth values then get blurred across all those pixels with a depth above a certain threshold, thus approximating out-of-focus objects behind the focal plane. The DOF technique proposed in this paper additionally considers the blurring of foreground objects and trivially handles semi-transparent regions.

3. Depth of Field

The depth of field is the *in-focus* range of a scene acquired by a camera with sufficient sharpness, in contrast to the *out-of-focus* regions of a scene, which appear gradually blurred in an image. This effect is due to the focal length and aperture of the lens. A lens allows light to pass through it and focuses the light rays onto an imaging plane, e.g. the retina of the eye, or the CCD sensor of a camera. For objects *in-focus*, the light rays will be directed to a single point on the plane while for *out-of-focus* objects, the light rays will be mapped to a circle-like area on the plane, called the *circle of confusion*, and thus appear blurred. Consequently, for objects to be in focus on the plane, they need to be at a certain distance from the lens and this relationship can be characterized by the thin lens equation (Equation 1 and Figure 2(a)), where f is the focal length, s is the distance from the lens to the image plane and z_f is the distance of an *in-focus* object at the focus plane to the lens [Ray94].

$$\frac{1}{f} = \frac{1}{z_f} + \frac{1}{s} \tag{1}$$

The *circle of confusion*, derived using basic geometry as illustrated in Figure 2(a) and Equation 2, describes the shape and size of the blur spot on the image plane, where $c(z)$ is the diameter of the *circle of confusion*, A is the aperture's diameter and z is the distance from the object to the lens.

$$c(z) = \left| A \frac{f(z_f - z)}{z(z_f - f)} \right| \tag{2}$$

Figure 2(b) shows the shape of the image space circle of confusion $c(z)$ as a function of the view space distance z. There, one can see that the size of the circle of confusion is not symmetric with respect to the focal place. Objects between the lens and the focal plane have a much bigger circle of confusion compared to those after the lens and thus appear more blurred. The circle of confusion converges to a constant value when the distance z goes to infinity.

An object is in focus if its circle of confusion is below the pixel size of the image or sensor c_{sharp}; it is then possible to compute [Ray94] the near and far depth of field limits $[D_{near}, D_{far}]$ for such a given acceptable circle of confusion c_{sharp}, as outlined in Equations 3 and 4:

$$D_{near} = \frac{A f z_f}{A f + c_{sharp}(f - z_f)} \tag{3}$$

$$D_{far} = \frac{A f z_f}{A f - c_{sharp}(f - z_f)} \tag{4}$$

$$z_h = \frac{(A + c_{sharp}) f}{c_{sharp}} \tag{5}$$

Such an acceptable circle of confusion allows also to solve for the so called *hyperfocal* distance z_h (Equation 5), beyond which all objects in a scene will appear in focus [Der06], since their circles of confusion are smaller than the pixels of the image. This effect allows the acquisition of images with a large depth range, such as in landscape photography, which is in contrast to macro-photography where scenes have a small depth range and as such the out-of-focus blur is a dominant phenomenon.

The circle of confusion can also be expressed in view space, as Equations 6, 7 and 8 show. The projection distance s is a function of the lens specific focal length f and the focused distance z_f. Together, they determine the magnification m of the projected image on the image plane.

$$C(z) = A \frac{|z_f - z|}{z} \tag{6}$$

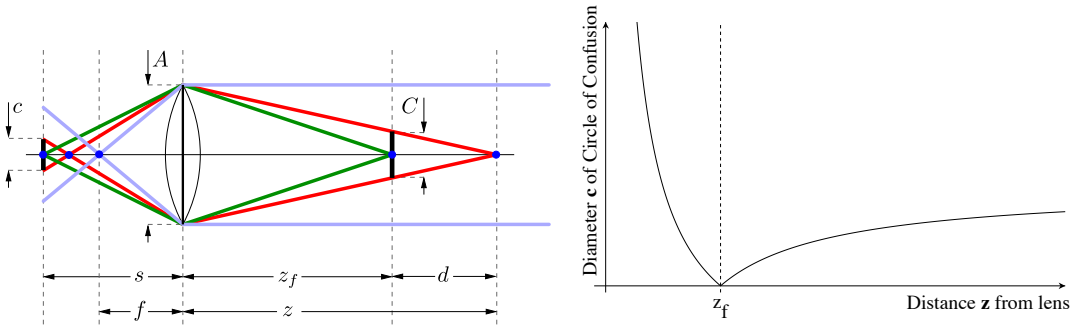
$$m = \frac{f}{z_f - f} \tag{7}$$

$$c(z) = C(z)\, m \tag{8}$$

Depth of field techniques have been successfully used by photo and film artists to create depth of field effects to guide a viewers gaze onto interesting parts of a scene. They have also been used to improve viewer immersion of synthetically generated images, often with focus on real-time entertainment or offline movie rendering. The following Section 3.1 motivates the presented method, followed by Sections 3.2 and 3.3, which discuss how depth of field effects can be integrated into a slice-based direct volume renderer to allow visualization of volumetric data sets.

3.1. Motivation of the Presented Approach

There are various approaches of integrating depth of field effects into a (volume) rendering system using generic methods described in the literature, such as rendering the scene from multiple slightly offset viewpoints [HA90] and accumulating the resulting images, approaching a physically correct rendering of depth of field when a sufficient number of



(a) geometric setup (b) image space circle of confusion

Figure 2: *a) Illustration of the basic geometric setup for the depth of field scene with the lens and rays for calculating the view space circle of confusion C of an unfocused object at distance z, when the lens is focused at the distance z_f. b) Illustration of the diameter of the circle of confusion $c(z)$ in image space as a function of the distance z of an object to the lens with the focal plane at z_f.*

viewpoints are combined. This however comes with a substantial performance penalty, since it requires traversing and compositing the volume once for each chosen viewpoint. For substantial depth of field effects, many of those viewpoints are required in order to reduce the "ghosting" of considerably blurred objects in the scene in the foreground and background, due to their a large circles of confusion [Dem04].

The layer-based method [Sco92, KS07] naturally maps to a slice-based volume rendering system by partitioning the scene along the slices required for volumetric traversal. Slices are then rendered and blurred individually and composited in either back-to-front or front-to-back order. Although it allows for a single-stack traversal, it exhibits an image quality/performance behavior similar to the method based on accumulating multiple viewpoints, since slices with a large circle of confusion require blurring across many samples in order to reduce aliasing and ghosting artifacts, necessitating multiple evaluations of the volume, transfer function and shading model.

The method presented in the following section is in spirit similar to the layer based method, however instead of *filtering each slice individually* and then *compositing* them, it *incrementally filters* an already *composited image* and then composits each subsequent slice with the already composited image. While necessitating a two-stack traversal, this avoids the performance penalty of accessing the volume, transfer function and evaluating the shading model more than once for each sample.

Image-space methods for adding depth of field effects to scenes consisting of solid polygons often use a z-buffer to estimate the circle of confusion, which is subsequently used to selectively blur the rendered image [Dem04]. That family of depth of field methods provides a good performance/image quality trade off, making it very suitable to applications in the entertainment industry, where solid geometry is the dominant type of workload encountered; it is however unclear how that approach can be robustly applied to direct volume rendering, where semi-transparent, potentially overlapping structures are common.

Ropinski *et al.* [RSH06] apply this idea to direct volume rendering by deriving a *pseudo* depth buffer by storing in a buffer the distances at which the accumulated opacities of rays cast reach a certain user specified threshold. The rendered color image is then blurred by approximating the circle of confusion based on the distances stored, but only for those pixels behind the focal plane, thus only partially approximating depth of field effects.

This opacity thresholding implies that some sort of solid features exist in the volume, and only those features are considered during the computation of the depth of field approximation. However, many data sets, e.g. those arising in the field of combustion simulation, often do not contain clear surface-like features since they store physical quantities, where transfer functions with smoothly varying colors and transparencies are often used to highlight ranges of the respective physical quantity.

Existing image-space methods based on opacity thresholding are not readily applicable to data sets containing transparent structures, and as such we present a method to render depth of field effects in a way that handles transparent regions trivially, without assuming intrinsic opaque surface-like features, thus making it especially applicable to rendering of data sets representing physical quantities of CFD and combustion simulations.

The following section presents the proposed method in detail and discusses design choices made in order to achieve

an easy integration of plausible depth of fields effects into a slice-based volume rendering system.

3.2. Integration into a Slice-Based Volume Renderer

Depth of field effects, as shown in Algorithms 1 and 2, can be interactively computed by modifying the traversal of the slices and their compositing of a slice-based volume renderer [EHK*06], which renders proxy geometry, typically slices created by intersecting view aligned planes with the bounding box of the volume on graphics hardware. A fragment program then computes color and opacity values to be composited with the image of previous slices already stored in the frame buffer.

There, CPU-computed proxy geometry, namely view-aligned slices, are first partitioned into those in front and those behind the focal plane of the lens, as illustrated in Figure 3(a). Each set is then traversed individually and composited into separate buffers, namely the slice-front buffer and the slice-back buffer, which both get subsequently blended together after their respective traversal.

The slices in front of the focal plane are projected into the slice-front buffer and traversed in front-to-back order during which the blurring introduced by the depth of field is approximated by averaging and compositing multiple samples from the previous slice-front buffer. The current and next slice-front buffers are swapped and the algorithm proceeds to the next slice.

Similarly, the slices behind the focal slice are traversed in back-to-front order and composited into the slice-back buffer during which multiple samples are also taken and averaged from the previous slice-back buffer, subsequently followed by swapping the current and next slice-back buffers before proceeding with the next slice. The process of averaging samples from the slice-front and slice-back buffers is detailed in Section 3.3.

The slice-front buffer is then blended on top of the slice-back buffer using the over operator, and the results are then copied into the frame buffer of the visible window, converting the final values from floating-point format to the fixed-point format usually used for displaying images on the screen. If required, tone-mapping can be applied.

3.3. Incremental Filtering for Computing Depth of Field

Blurring across the circle of confusion, as described in Equation 6 and illustrated in Figure 3(a), is done by averaging samples from the previous slice-front/slice-back buffer in a neighborhood around the current fragment when updating the current slice-front/slice-back buffer.

Evaluating the image space circle of confusion $c(z)$ directly as a function of the view space distance z (Equations 8 and 7, Figure 2(b)) is equivalent to performing a perspective projection, which in interactive volume rendering systems is commonly done by transforming points by a perspective projection matrix in combination with the homogeneous divide performed by the graphics hardware.

A circle of confusion at distance z with view-space diameter $C(z)$ projects to a circle in the image plane when viewed from the camera's point of view, which scales in image space as a function of the view space distance z. This scaling is described by the (perspective) projection matrix P which can potentially non-uniformly scale objects, e.g. if the aspect ratio of the projection matrix does not match that of the viewport, thus transforming the circle of confusion from view space into an ellipse in the image plane, which is implicitly considered during the following derivations. The results presented in this paper however did not exhibit projections with non-uniform scaling in image-space.

The view space diameter of a circle of confusion $C(z)$ at view space distance z is represented by a homogeneous point $C_{view} = (C(z)\ C(z)\ z\ 1)^T$ which is projected from view space into clip space by the projection matrix P. The xy coordinates of points in clip space are within the $[-1,\ 1]^2$ interval; texture coordinates typically used to access images using graphics hardware are in the range of $[0,\ 1]^2$. Therefore, a scale and bias matrix T, as shown in Equation 9 is required which maps points from clip space to the texture coordinate space $\vec{C}_c$, which is summarized in Equation 10. The final step is performing the perspective division to yield the projected texture space circle of confusion $\vec{C}_t$, shown in Equation 11:

$$T = \begin{pmatrix} 0.5 & 0 & 0 & 0.5 \\ 0 & 0.5 & 0 & 0.5 \\ 0 & 0 & 0.5 & 0.5 \\ 0 & 0 & 0 & 1.0 \end{pmatrix} \tag{9}$$

$$\vec{C}_c = \begin{pmatrix} x_t \\ y_t \\ z_t \\ w_t \end{pmatrix} = T \cdot P \cdot \begin{pmatrix} C(z) \\ C(z) \\ z \\ 1 \end{pmatrix} \tag{10}$$

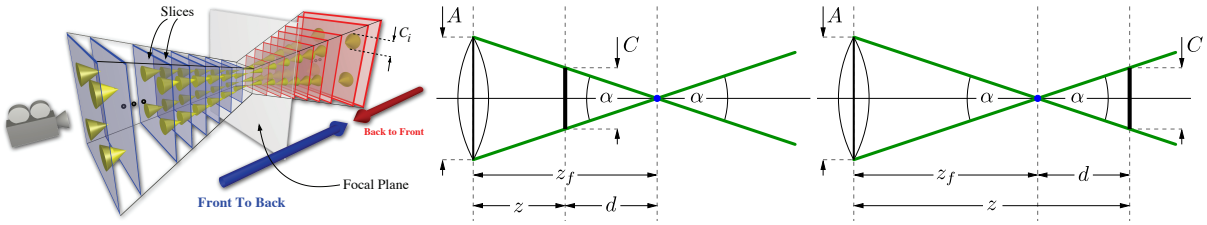
$$\vec{C}_t = \begin{pmatrix} \frac{x_t}{w_t} \\ \frac{y_t}{w_t} \end{pmatrix} \tag{11}$$

The texture space circle of confusion $\vec{C}_t$ is passed to the fragment shader where it is used to compute texture coordinate offsets for sampling the previous slice-front/slice-back buffer. A set of N sample offsets $\vec{p}_i$ is generated on the circle/ellipse with extent $\vec{C}_t$ and added to the projected texture coordinate $\vec{f}_t$ of the currently processed fragment, yielding the set of texture space sample positions $\vec{t}_i$, as Equation 12 shows:

$$\vec{t}_i = \vec{f}_t + \vec{p}_i \tag{12}$$

Sample offsets $\vec{p}_i$ were initially created using a Poisson distribution, but experimentation suggested a regular grid of user specified resolution (e.g. $N = 4$, for a 2×2 grid) as a reasonable compromise between image quality and performance.

The previous slice-front/slice-back buffer is then sampled



(a) Geometric setup (b) Slice in front of the focal plane z_f (c) Slice behind the focal plane z_f

Figure 3: *a) Geometric setup of computing depth of field effects by traversing the slices of a direct volume rendering system in two separate passes; those in front of the focal plane in front-to-back order and those behind the focal plane in back-to-front order. The bases of the yellow cones denote the regions from which samples are taken from the previous buffer during the incremental filtering. Slices are scaled in screen space to guarantee that the contribution of a slice is correctly considered by the incremental filtering of subsequent slices, due to linearly increasing circles of confusion C_i, which are 0 at the slice closest to the focal distance z_f. b) The proposed simplified depth of field model with the user specified parameter α describing the rate of change with which the circle of confusion changes in view space as a function of the distance d to the focal plane situated at z_f for a slice at distance z in front of the focal plane z_f and b) for a slice at distance z behind the focal plane z_f.*

at all sample positions $\vec{t}_i$ and those are accumulated and divided by the total number of samples N. This average is then taken as the *destination* parameter in the front-to-back compositing equation for the current slice-front buffer, or the back-to-front composition equation for the current slice-back buffer respectively. The *source* parameter is the result of evaluating the 2D scalar/gradient-magnitude transfer function and a shading model, such as the commonly used emission-absorption model or the Phong surface shading model.

In typical slice-based volume rendering systems, proxy geometry is generated by intersecting slicing planes against the bounding box of the data set, however the increase of C as a function of d requires scaling the slice in screen space by C to guarantee that the contribution of a slice is correctly considered by the incremental filtering of subsequent slices. Figure 3(a) illustrates the required proxy geometry, namely two frustums joined together with their top surfaces at the focal plane, their taper being a function of the circle of confusion C.

Empirically we observed that it was rather difficult and impractical for a user to manipulate the lens parameters (Equation 2) directly, in order to achieve a specific visualization. As such a more intuitive method was derived, based on the observation that the circle of confusion is proportional in view space to the difference $d = |z_f - z|$ between the object distance z and the focal distance z_f, which describes depth of field *qualitatively*, whereas the lens properties describe it in a *quantitative* way. The results presented were created with a simplified model to specify the depth of field parameters, as illustrated in Figure 3 and outlined in Equation 13.

$$C(d) = 2\tan\left(\frac{\alpha}{2}\right)d \tag{13}$$

$$A(z_f) = 2\tan\left(\frac{\alpha}{2}\right)z_f \tag{14}$$

The camera always focuses on a user specified point in model space, and as such, the focal distance z_f is changing in view space when the user changes the camera position or orientation. The angle α then determines the rate $\tan\left(\frac{\alpha}{2}\right)$ with which the circle of confusion increases as a function of the distance d. The lens aperture A, shown in Equation 14 is then a function of z_f and α.

This frees the user from the burden of refocusing z_f when the camera is moved, e.g. when zooming into a specific region of the data set, and also allows quick shifts of focus, e.g. from the front of the volume to the back, irrespective of where the camera is located. Specifying α directly instead of A keeps the region in focus at a constant range, independent of the focus distance.

4. Results and Discussion

The method was implemented using OpenGL and Cg running on a mid-range NVIDIA GeForce GTX 460 GPU with 1 GB of video memory. The images were rendered at a 512×512 resolution into 16-bit precision floating-point buffers with an inter-slice distance of $d = 0.001$ to capture high-frequency details introduced by multi-dimensional transfer functions and unless noted otherwise, depth of field effects were rendered by taking 2×2 samples when blurring during slice traversal.

The backpack CT data set is shown in Figure 1, where the camera is focusing on objects at different depths, such as the spray can in the foreground, the wires behind, and the

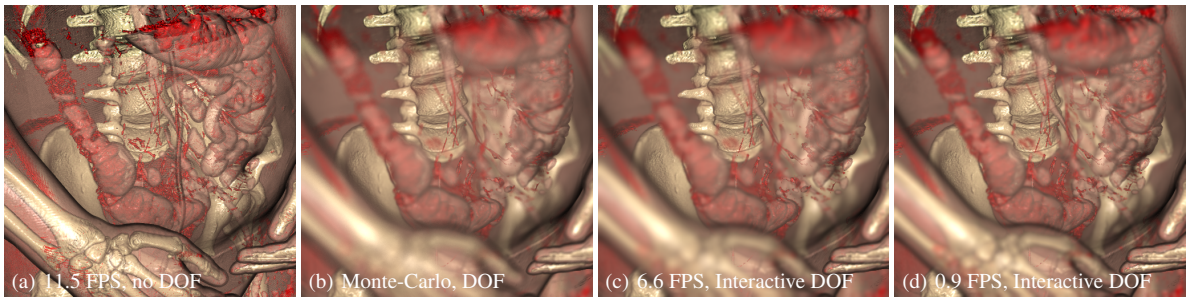


Figure 4: *The abdomen part of the visible human data set with* $512 \times 512 \times 512$ *voxels rendered with a) Phong shading and b) a depth of field effect created by an offline Monte-Carlo raytracer c) a depth of field effect with the presented interactive method with* $\alpha = 44.6^\circ$ *and* 2×2 *samples taken during the incremental filtering and d) which is similar to image c), but here, 81 samples following a Poisson distribution were used during the incremental filtering. The focus is on the spine, the skin is rendered semi-transparently for additional context. In the images, 853 slices were taken and gradients were computed on the fly.*

boxes and the other spray can in the background, as shown in Figures 1(b) to 1(d). The blurred out-of-focus objects introduced by the depth of field approximation make it easier to intuitively reason about the spatial arrangement of in-focus objects. Noticeable is also the increased ability to locate the small circular objects distributed throughout the scene, compared to Figure 1(a).

Figure 4 shows the abdomen of the Visible Human CT data set where the bones and parts of the internal organs were classified using the transfer function. The lack of depth of field, as shown in Figure 4(a), yields significant visual clutter, which is drastically reduced in Figure 4(c) with the camera focused on the spine. The blurred internal organs and the semi-transparent skin layer support the comprehension of the relative arrangement of the features contained in the data set.

Figure 5 shows the Richtmyer-Meshkov fluid data set rendered with Phong shading in Figure 5(a). The high frequencies of the features contained in this fluid simulation data set makes it relatively hard to make out distinct features, it is difficult to get an intuition how those eddies are arranged with respect to each other. Figure 5(b) shows the same scene, but this time with the camera focused on an eddy in the foreground. Figure 5(c) then emphasizes the prominent yellow eddy in the center with the surrounding features gradually becoming out-of-focus. Moving the focus plane behind the yellow eddy, as shown in Figure 5(d), guides the gaze of the observer towards the other yellow eddies at the boundary of the data set, shown in the far back. The interactive depth of field approximation is able to point out features at different distances, while maintaining a global impression of the whole data set.

Figure 6 shows the aneurysm MRI data set rendered without depth of field in Figure 6(a) and with depth of field effects with focus on the enlarged blood vessel. The circle of confusion rate of change is increased from $\alpha = 57.6^\circ$ in Figure 6(b) to $\alpha = 161.2^\circ$ in Figure 6(c), causing features relatively far away from the focus plane to be blurred more, thus expanding their screen space coverage. Features closer towards the focus plane are less affected, especially when their respective image space circles of confusion are smaller than the pixel size of the blurred buffers. Increasing the number of samples taken during blurring changes the shape of the blur subtly (Figure 6(c) and Figure 6(d)), especially for regions that are strongly defocused, at the price of a considerable performance drop.

Low values of α create more subtle depth of field effects, putting less emphasis on the regions in focus, while large values for α tend to deemphasize the surrounding features. Values of $\alpha \in [25^\circ, 100^\circ]$ provide reasonable depth of field effects for a variety of data sets, as demonstrated in the presented results, and can be used as a starting point for further experimentations to adjust the strength of depth of field effects for specific applications.

The incremental filtering of an already blurred and composited buffer decomposes into filtering with a large filter width, e.g. at the slice closest to the viewer with commonly large circles of confusion, into many filtering operations with much smaller filter widths, typically requiring only small numbers of samples, since they cover only small regions of the already blurred and composited buffer. This is in contrast to a direct implementation of a layer-based method [Sco92, KS07], where out-of-focus slices require the number of samples being proportional to the potentially large circles of confusion in order to yield blurred images of sufficient visual quality.

Notable is that the incremental filtering of an already blurred and composited buffer causes a much larger *effective* blur, since the contribution of the first slice will be blurred again for each subsequent slice. This increased amount of

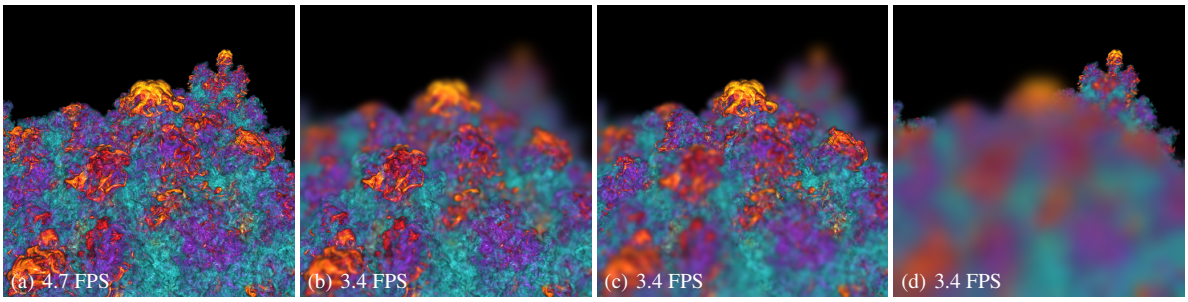


Figure 5: *A* $1024 \times 1024 \times 384$ *voxel subset of the Richtmyer-Meshkov data set rendered with a) Phong shading and subsequently DOF with* $\alpha = 105.7^\circ$ *and focus on b) an eddy in the front, c) the center eddy and d) the far yellow eddy. In each image 1350 slices were taken and gradients were computed on the fly due to memory constraints.*

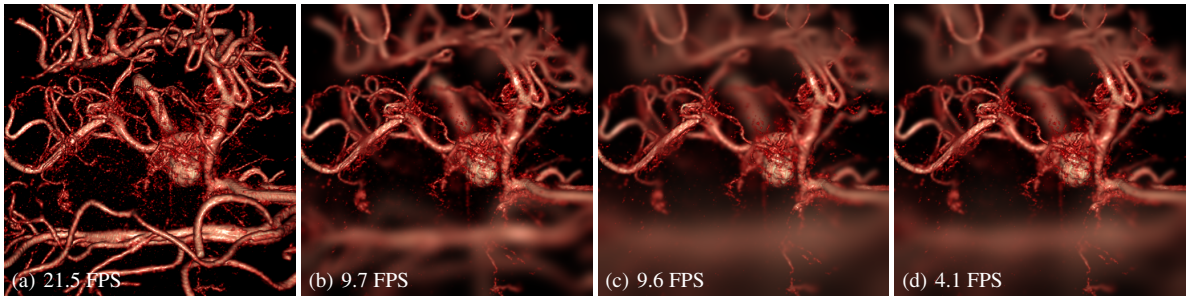


Figure 6: *The aneurysm data set with* $256 \times 256 \times 256$ *voxels rendered with Phong shading and a) no depth of field, b) depth of field with focus on the damaged blood vessel and* $\alpha = 57.6^\circ$ *and c) focused at the same plane but a stronger depth of field effect by setting* $\alpha = 161.2^\circ$*, and d) which is similar to the previous image, but a sample grid resolution of* 4×4 *compared to the* 2×2 *grid used in the previous images. For each image, 655 slices were taken and gradients were precomputed and fetched from a 3D texture during rendering.*

blur however does not significantly impair the ability of an observer to qualitatively reason about the depth of a scene, since the human visual system is rather ineffective at using various *levels* of blur to aid in depth perception, as discussed by Mather *et al.* [MS02].

Figure 4(c) demonstrates that the proposed method creates qualitatively similar looking images compared to the depth of field effects created by a Monte-Carlo raytracer, shown in Figure 4(b), where α was chosen empirically to match the physical lens parameters of the ray traced image. Notable differences are the aforementioned increased blurriness of out of focus regions due to the incremental filtering, which can be slightly reduced by increasing the number of samples taken during the blurring, as Figure 4(d) shows.

The presented incremental filtering method necessitates two stacks of slices which are individually traversed such that the blur filter widths of slices *decrease* in traversal order, thus making sure that the *effective* blur of slices close to the focal plane is lower than that of the out-of-focus slices of the foreground or background respectively.

Additional slice stacks could also be used to support multiple depth of field focus planes where the blur filter width is either increasing or decreasing for a specific stack, thus allowing an extension of Kosara's *Semantic Depth of Field* to volume rendering [KMH01, Kos01]. However the constraints of the traversal order make it less trivial to integrate advanced shading models which depend on a consistent front-to-back slice traversal across the whole slice stack [KPH*03, SPH*09, vPBV10].

The presented method, in contrast to other methods, such as those based on illustrative rendering [ER00], does not attempt to abstract features and separate them into foreground and background objects, which instead, as Figures 1 and 5 show, are indirectly separated by the blurring based on the distance to the focal plane, which is orthogonal to the specific shading method used.

The proposed method enables an easy way to add plausible depth of field effects using intuitive user specified parameters, it however does not capture all aspects of a physical lens. For example, a physical lens focused beyond its

hyperfocal distance will extend the far depth of field limit to infinity, thus rendering distant objects in focus. The approximation described here however does not exhibit this behavior, since the effective blur of slices increases, the further they are behind the focal plane. This is rarely a problem in practice though, since volumetric data sets typically extend over a relatively small depth range, thus most of the volume is close with respect to the focal plane, where the quantitative differences with respect to a physical lens are minimal. Scenes in landscape photography however encompass large depth ranges, and as such correctly handling of objects beyond the hyperfocal distance is critical, whereas volume rendering of CT and MRI data sets is essentially macro photography where the depth of field is small.

5. Conclusion and Future Work

In this paper, a method has been presented that adds depth of field effects to direct volume rendering, which were shown to aid in the inference of depth cues. The method does not rely on pre-computation and therefore allows interactive manipulation of camera position and transfer function, for both solid and transparent regions.

Incorporating the proposed method into an existing slice-based direct volume rendering system required minimal modifications and allowed the rendering of depth of field effects with frames rate reduced by a factor less than 50%. Performance can potentially be increased by changing the number of samples during incremental filtering as a function of the texel space circle of confusion.

In the future, we would like to extend the proposed technique to handle tilt shifting effects and chromatic aberration, which change the circle of confusion as a function of a tilted focal plane and the wavelength of light respectively. This, combined with compensating for the over blurring due to the incremental filtering, could be used to add physically correct depth of field effects to an interactive preview tool of an offline realistic rendering system which renders volumetric primitives using a more physically correct camera model.

It would also be interesting to quantify the perceptual gains of depth of field effects and their interplay with different local and global shading models in the context of direct volume rendering and visualization.

Acknowledgements

We would like to thank the Computer Graphics Groups of the Universities of Vienna and Erlangen as well as the Lawrence Livermore National Lab for providing the data sets and the reviewers for their thoughtful comments and suggestions.

This publication is based on work supported by he National Nuclear Security Administration under the Accelerating Development of Retrofitable CO2 Capture Technologies through Predictivity through DOE Research Grant #DE-NA0000740 and Award No. KUS-C1-016-04, made by King Abdullah University of Science and Technology (KAUST), and DOE VACET SciDAC, NSF OCI-0906379, NSF CNS-0615194.

Appendix A: Pseudocode for the depth of field algorithm

Algorithm 1 The main rendering algorithm

```
ComputeProxyGeometry
fp ⇐ ComputeSliceIndexOfFocalPlane()
slice_front_buffer_prev ⇐ slice_front_buffer_next ⇐ 0
for slice s = 0 to fp do
  Blur(slice_front_buffer_prev, slice_front_buffer_next,
  s, FRONT_TO_BACK)
end for
slice_back_buffer_prev ⇐ slice_back_buffer_next ⇐ 0
for slice s = s_max to fp do
  Blur(slice_back_buffer_prev, slice_back_buffer_next,
  s, BACK_TO_FRONT)
end for
CompositWithWindowFrameBuffer(slice_back_buffer)
CompositWithWindowFrameBuffer(slice_front_buffer)
```

Algorithm 2 Blur(next_buffer, previous_buffer, s, traversal_direction)

```
BindTexture(previous_buffer)
SetRenderTarget(next_buffer)
DisableBlending()
DrawCurrentSlice()
d ⇐ slice_distance · |s − fp|
C ⇐ 2 · d · tan(α/2)
compute C_t in the vertex shader as in Equation 8
for all fragments f of current slice do
  (x, |∇x|) ⇐ Texture3D(volume, f)
  (color_rgb, σ_t) ⇐ Texture2D (transfer_func, (x, |∇x|))
  shaded_rgb = ShadeSample (color_rgb)
  shaded_a = 1 − e^(−σ_t · slice_distance)
  blurred_rgba ⇐ 0
  for all samples p_i within C_t do
    compute t_i as in Equation 12
    blurred_rgba += Texture2D (previous_buffer, t_i)
  end for
  blurred_rgba ⇐ blurred_rgba / N
  source ⇐ shaded · shaded_a
  destination ⇐ blurred
  if traversal_direction == BACK_TO_FRONT then
    composited ⇐ source + destination · (1 − source_a)
  else
    composited ⇐ source · (1 − destination_a) + destination
  end if
end for
next_buffer ⇐ composited
Swap(previous_buffer, next_buffer)
```

References

[BHK*03a] BARSKY B. A., HORN D. R., KLEIN S. A., PANG J. A., YU M.: Camera models and optical systems used in computer graphics: Part i, object based techniques. In *Computational Science and Its Applications - ICCSA 2003*, Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2003, pp. 246–255. 2

[BHK*03b] BARSKY B. A., HORN D. R., KLEIN S. A., PANG J. A., YU M.: Camera models and optical systems used in computer graphics: Part ii, image-based techniques. In *Computational Science and Its Applications - ICCSA 2003*, Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2003, pp. 256–265. 2

[BK08] BARSKY B. A., KOSLOFF T. J.: Algorithms for rendering depth of field effects in computer graphics. In *Proceedings of the 12th WSEAS international conference on Computers* (Stevens Point, Wisconsin, USA, 2008), World Scientific and Engineering Academy and Society (WSEAS), pp. 999–1010. 2

[CNL*09] CRASSIN C., NEYRET F., LEFEBVRE S., SAINZ M., EISEMANN E.: Beyond triangles: gigavoxels effects in video games. In *SIGGRAPH 2009: Talks* (2009), SIGGRAPH '09, ACM, pp. 78:1–78:1. 2

[CNLE09] CRASSIN C., NEYRET F., LEFEBVRE S., EISEMANN E.: Gigavoxels: ray-guided streaming for efficient and detailed voxel rendering. In *Proceedings of the 2009 symposium on Interactive 3D graphics and games* (2009), I3D '09, ACM, pp. 15–22. 2

[CPC84] COOK R. L., PORTER T., CARPENTER L.: Distributed ray tracing. *SIGGRAPH Comput. Graph. 18* (January 1984), 137–145. 2

[CWV07] CIUFFREDA K. J., WANG B., VASUDEVAN B.: Conceptual model of human blur perception. *Vision Research 47*, 9 (2007), 1245 – 1252. 2

[Dem04] DEMERS J.: *GPU Gems*. Addison-Wesley Longman, Inc., 2004, ch. 23. Depth of Field: A Survey of Techniques, pp. 375 – 390. 2, 4

[Der06] DERR L.: *Photography for students of physics and chemistry*. Macmillan & Co., Ltd, 1906, ch. 6, p. 79. 3

[EHK*06] ENGEL K., HADWIGER M., KNISS J., REZK-SALAMA C., WEISKOPF D.: *Real-Time Volume Graphics*. A K Peters, July 2006. 5

[ER00] EBERT D., RHEINGANS P.: Volume illustration: nonphotorealistic rendering of volume models. In *Proceedings of the conference on Visualization '00* (Los Alamitos, CA, USA, 2000), VIS '00, IEEE Computer Society Press, pp. 195–202. 8

[HA90] HAEBERLI P., AKELEY K.: The accumulation buffer: hardware support for high-quality rendering. *SIGGRAPH Comput. Graph. 24* (September 1990), 309–318. 3

[HCOB10] HELD R. T., COOPER E. A., O'BRIEN J. F., BANKS M. S.: Using blur to affect perceived distance and size. *ACM Transactions on Graphics 29*, 2 (Mar. 2010), 1–16. 1, 2

[KMH01] KOSARA R., MIKSCH S., HAUSER H.: Semantic depth of field. *IEEE Symposium on Information Visualization, 2001. INFOVIS 2001.* (2001), 97–104. 2, 8

[KMH*02] KOSARA R., MIKSCH S., HAUSER H., SCHRAMMEL J., GILLER V., TSCHELIGI M.: Useful properties of semantic depth of field for better f+c visualization. In *Proceedings of the symposium on Data Visualisation 2002* (Aire-la-Ville, Switzerland, Switzerland, 2002), VISSYM '02, Eurographics Association, pp. 205–210. 2

[Kos01] KOSARA R.: *Semantic Depth of Field - Using Blur for Focus+Context Visualization*. PhD thesis, Vienna University of Technology, Vienna, Austria, 2001. 2, 8

[KPH*03] KNISS J., PREMOZE S., HANSEN C., SHIRLEY P., MCPHERSON A.: A model for volume lighting and modeling. *IEEE Transactions on Visualization and Computer Graphics 9*, 2 (2003), 150–162. 2, 8

[KS07] KRAUS M., STRENGERT M.: Depth-of-field rendering by pyramidal image processing. *Computer Graphics Forum 26* (September 2007), 645–654. 2, 4, 7

[MS02] MATHER G., SMITH D. R. R.: Blur discrimination and its relation to blur-mediated depth perception. *Perception 31*, 10 (2002), 1211–1219. 2, 7

[MS04] MATHER G., SMITH D. R. R.: Combining depth cues: effects upon accuracy and speed of performance in a depth-ordering task. *Vision Research 44*, 6 (2004), 557 – 562. 2

[NSG90] NOVINS K. L., SILLION F. X., GREENBERG D. P.: An efficient method for volume rendering using perspective projection. *SIGGRAPH Comput. Graph. 24* (November 1990), 95–102. 2

[PC81] POTMESIL M., CHAKRAVARTY I.: A lens and aperture camera model for synthetic image generation. In *Proceedings of the 8th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1981), SIGGRAPH '81, ACM, pp. 297–305. 2

[Ray94] RAY S. F.: *Applied Photographic Optics: Lenses and Optical Systems for Photography, Film, Video and Electronic Imaging*. Focal Press, 1994. 3

[RBGV08] RAUTEK P., BRUCKNER S., GRÖLLER E., VIOLA I.: Illustrative visualization: new technology or useless tautology? *SIGGRAPH Comput. Graph. 42* (August 2008), 4:1–4:8. 2

[Rok96] ROKITA P.: Generating depth-of-field effects in virtual reality applications. *IEEE Computer Graphics and Applications 16* (1996), 18–21. 2

[RSH06] ROPINSKI T., STEINICKE F., HINRICHS K.: Visually supporting depth perception in angiography imaging. In *Smart Graphics* (2006), Springer, pp. 93–104. 2, 3, 4

[Sco92] SCOFIELD C.: *2 1/2-D depth-of-field simulation for computer animation*. Academic Press Professional, Inc., San Diego, CA, USA, 1992, pp. 36–38. 2, 4, 7

[SPH*09] SCHOTT M., PEGORARO V., HANSEN C., BOULANGER K., BOUATOUCH K.: A directional occlusion shading model for interactive direct volume rendering. In *Computer Graphics Forum (Proceedings of Eurographics/IEEE VGTC Symposium on Visualization 2009)* (2009), vol. 28. 2, 8

[VKG04] VIOLA I., KANITSAR A., GROLLER M.: Importance-driven volume rendering. In *Visualization, 2004. IEEE* (2004), pp. 139 – 145. 2

[vPBV10] ŠOLTÉSZOVÁ V., PATEL D., BRUCKNER S., VIOLA I.: A multidirectional occlusion shading model for direct volume rendering. *Computer Graphics Forum 29*, 3 (J 2010), 883–891. 2, 8

[WWLM11] WANG Y.-S., WANG C., LEE T.-Y., MA K.-L.: Feature-preserving volume data reduction and focus+context visualization. *Visualization and Computer Graphics, IEEE Transactions on 17*, 2 (2011), 171 –181. 2

[WZMK05] WANG L., ZHAO Y., MUELLER K., KAUFMAN A.: The magic volume lens: an interactive focus+context technique for volume rendering. In *Visualization, 2005. VIS 05. IEEE* (2005), pp. 367 – 374. 2